FIELD OF THE INVENTION

The invention relates to a video apparatus, notably a video decoder, and to a process for controlling such an apparatus.

BACKGROUND OF THE INVENTION

It is known to provide a video apparatus with a decoder circuit, for instance a MPEG decoder, in order to generate a video signal usable by a display, for instance as a CVBS signal or as a RGB signals, from a video digital stream. Such a decoder circuit uses a so-called Video RAM (random-access-memory) to retain data which are processed, for instance to decompress a MPEG stream.

Generally, a video apparatus also comprises an OSD circuit (OSD stands for On-Screen Display) to generate and send to the display images to be superimposed on the video sequence output by the decoder circuit ; these images are often menus with graphics.

The OSD circuit also needs RAM to generate and process the graphics, i.e. the OSD images.

SUMMARY OF THE INVENTION

The invention seeks to provide a video apparatus with a decoder circuit and an OSD circuit with memory architecture with reduced memory size and thus also proposes a process to control this memory architecture according to the mode of operation of the video apparatus.

The invention proposes a video apparatus with a digital decoder having a first memory on an internal bus and linked to an OSD circuit and to a second memory via a main bus, the video apparatus comprising means for realising a DMA transfer between the first memory and the second memory.

Preferably, a CPU is connected to the main bus and the second memory is used by the CPU. Possibly the first memory is a Video RAM and wherein the second memory is a CPU RAM.

In a possible embodiment, the digital decoder is connected to a digital front-end.

The invention also proposes a process for controlling a video apparatus with a digital decoder having a first memory on an internal bus and linked to an OSD circuit and to a second memory via a main bus, comprising the step of realising a DMA transfer between the first memory and the second memory via the digital decoder.

The following steps are proposed :

- issuing a request for the OSD circuit to use more than a given size in the second memory,
- realising a DMA transfer from the second memory to the first memory

The following further steps are also possible :

- issuing a request for the OSD circuit to use data in the first memory,
- copying via a DMA transfer data from the second memory to the first memory ;
- realising a DMA transfer of the requested data from the first memory to the second memory.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be explained with reference to figure 1 representing a video apparatus according to the invention. The video apparatus of figure 1 is a satellite decoder 2. Only the parts which are necessary for understanding the invention have been represented.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

An antenna 3 receives a signal representing at least one video sequence from a satellite. An input pin of the decoder 2 receives the signal transmitted by the antenna 3 to forward it to a digital front-end 4 comprising notably a tuner and a demodulator.

From the antenna signal, the digital front-end generates a MPEG stream which is converted to a CVBS signal by a MPEG decoder 6. To decompress the MPEG stream, the MPEG decoder 6 is connected via a data bus to a video RAM 8.

On the other hand, the satellite decoder 2 also has an OSD circuit 12 for generating, upon instructions from a CPU 14, images (called graphics hereafter) to be superimposed on the CVBS signal. The graphics to be displayed are coded in RGB on a Scart connector with a fast blanking signal FB indicating when points of the graphics have to be displayed.

The CPU 14 and the OSD circuit 12 share a RAM, called CPU RAM 10, via a common data bus 16. The MPEG decoder 6 is also connected on this common bus 16.

The Video RAM 8 and the CPU RAM 10 can exchange data on the common bus 16 through the MPEG decoder 6 by DMA (DMA stands for Direct Memory Access). It should be noted however that the Video RAM 8 is not directly accessible from the CPU 14.

The system has to cope with three different memory sizes available for OSD depending on the configuration (mode of operation) :

- Configuration 1 : Video displayed

When moving pictures are displayed, the desirable RAM minimum size available for OSD should allow to store 262144 pixels in CLUT4 (Colour Lock-Up Table where 1 pixel = 4 bits) mode, which requires 131072 bytes memory space.

- Configuration 2 : Still pictures displayed

When still pictures are displayed, the RAM size available for OSD can be extended to 996148 pixels in CLUT4 mode, which requires 448074 bytes memory space.

- Configuration 3 : no video nor still pictures

When no moving pictures and still pictures are displayed, all video RAM should preferably become available for OSD, which represents 1.9 MB memory space approximately.

The CPU RAM 10 has a 2MB (Mega Bytes) capacity.

The Software occupies 1.25 MB in CPU RAM 10, until the scheduler has started. 750 KB are then left available for the system and the OSD buffer pools. A 150 KB system pool is sufficient to insure a robust and efficient functioning of the software. It then remains around 600 KB in CPU RAM 10 for the OSD pool. When video is running (configuration 1), 112 KB of free memory space are available in Video RAM 8. Configuration 2 leaves 457 KB of available memory in Video RAM 8, whereas when no video and still pictures are running (configuration 3), almost the entire Video RAM 8 becomes available, which represents around 1.9 MB. In configuration 1 and 2, the 600 KB available memory in CPU RAM 10 are sufficient to cover the preferable OSD sizes stated above.

Configuration 3 demands 1.9 MB of memory, which is more than the 600 KB available in RAM CPU 10. In configuration 3, RAM CPU 10 contains both the buffers displayed in the buffers currently used in RAM CPU, which represents 2*207360 = 414720 bytes (2 full screen buffers in CLUT4 mode, one displayed, one being used). The other buffers are stored in RAM Video 8. When a buffer is no more displayed or used, it is flushed to Video RAM 8 via a DMA transfer. When a buffer stored in Video RAM 8 has to be displayed or comes in use, it is loaded in the CPU RAM by a DMA transfer. With this mechanism, the Video RAM 8 plays the role of a cache memory for the CPU RAM 10.

4 different states have thus been defined depending on the OSD configuration :

- State 1 : Video (Video running)
- State 2 : Still (still picture running)
- State 3 : OSD RAM CPU (Only OSD running within the limits of 622080 bytes allocated)
- State 4 : OSD RAM Video (Only OSD running with more than 622080 bytes allocated)

States 1, 2, and 3 correspond to the normal memory mapping, where all the OSD buffers are located in CPU RAM. State 4 correspond to the memory mapping where all the Video RAM is available for OSD buffers.

States 1, 2 and 3 are managed the same way by the driver of the OSD circuit 12, since all OSD buffers allocated will be placed in RAM CPU 10. Direct transitions

between state 1, 2 and state 4 can't happen, because in state 4, the Video RAM is used for the OSD and isn't available for still pictures or video. A transition to state 3 is compulsory before going to state 4. Therefore, the only time the OSD driver has to deal with Video RAM 8 concerns the transitions between state 3 and state 4.

Transition from state 3 to state 4 happens when the application asks the driver to create a new display by calling an OSD_credisplay function and when the total size allocated in CPU RAM 10 for the OSD displays (after the OSD_credisplay call) overflows the 622080 bytes available in CPU RAM 10. In this case, Video RAM 8 shall be activated. A pool of 1.9 MB shall then be created in Video RAM, all the OSD buffers stored in CPU RAM shall be transferred in RAM Video, the display descriptors updated accordingly, and the buffers displayed and the working buffer shall stay in CPU RAM.

Transition to state 4 to state 3 happens when the application asks the driver to free a display by calling a OSD_free_display function and when the total size allocated for the OSD displays (after the OSD_free_display call) becomes inferior to 498074 bytes (corresponding to the size needed in still picture mode). In this case, Video RAM 8 shall be deactivated and shall not be used anymore by the OSD driver. All the OSD buffers in Video RAM 8 shall then be transferred in CPU RAM 10, the display descriptor updated accordingly, and the pool in Video RAM 8 shall be deleted.

When in state 4, the Video RAM 8 is used as a cache for the OSD. The management of OSD regions and buffers use the same structures as the one already in use. The only difference is that the buffer address stored in the OSD buffer control blocks in CPU RAM 10 correspond to Video Ram 8 address in state 4, whereas the correspond to CPU RAM address in state 1, 2 and 3. To help management of the Video RAM 8 as cache, an internal array of structure is used, which contains the buffer ID, the address in CPU RAM 10, the address in Video RAM 8, the size and a pointer to the buffer descriptor, for each of the 16 display buffers and the buffers currently drawn.

Before drawing or displaying an OSD buffer placed in Video RAM 8, the driver will first have to transfer it from Video RAM 8 to CPU RAM 10. When a displayed

buffer or the currently drawn buffer isn't used anymore and is replaced by another one, the driver as to flush it in Video RAM (i.e. transfer it from CPU RAM 10 to Video RAM 8). In both cases, the array structure will be updated correspondingly.